מדינת ישראל
STATE OF ISRAEL

nistry of Justice
Patent Office

משרד המשפטים
לשכת הפטנטים

is to certify that annexed
is a true copy of the
ents as originally
ed with the patent
ion of which
rs are specified on the
of the annex.

זאת לתעודה כי רצופים
בזה העתקים נכונים של
המסמכים שהופקדו
לכתחילה עם הבקשה
לפטנט לפי הפרטים
הרשומים בעמוד הראשון
של הנספח.

CERTIFIED COPY OF PRIORITY DOCUMENT

This 22-01-2001 היום

רשם הפטנטים
Commissioner of Patents

נתאשר
Certified

חוק הפטנטים, תשכ"ז-1967
Patent Law, 5727 - 1967

בקשה לפטנט
Application for Patent

לשימוש הלישכה
For Office Use

מספר: 124197 Number

תאריך: 23-04-1998 Date

הוקדם/נדחה Ante/Post-Dated

אני, (שם המבקש, מענו ולגבי גוף מאוגד - מקום התאגדותו)
I, (Name and address of applicant, and in case of body corporate-place of incorporation)

DSPC Israel, Ltd.
11 Ben Gurion Street
Givat Shmuel 51905

די. אס. פי. סי. ישראל בע"מ
רח' בן גוריון 11
גבעת שמואל 51905

Inventor: המציא: 1. דוד בן-אלי
David BEN-ELI

בעל ההמצאה מכח ASSIGNMENT העברה שמה הוא
Owner, by virtue of an invention the title of which is

(בעברית)
(Hebrew) אפנן QPSK π/4

(באנגלית)
(English) π/4 QPSK MODULATOR

hereby apply for a patent to be granted to me in respect thereof. מבקש בזאת כי ינתן לי עליה פטנט

*בקשת חלוקה - Application of Division	*בקשת פטנט מוסף - Application for Patent Addition	*דרישה דין קדימה Priority Claim		
*מבקשת פטנט from Application	*לבקשה/לפטנט to Patent/Appl.	מספר/סימן Number/Mark	תאריך Date	מדינת האיגוד Convention Country
מס' Nº _____	מס' Nº _____			
מיום dated _____	מיום dated _____			
*יפוי כח: כללי / ~~מיוחד - רצוף בזה / עוד יוגש~~ P.O.A.: general / ~~individual - attached / to be filed later~~				
הוגש בענין filed in case 121843				
המען למסירת מסמכים בישראל Address for Service in Israel לוצאטו את לוצאטו ת.ד. 5352 באר שבע 84152 מספרנו: 4512/97				
חתימת המבקש Signature of Applicant *Luzzatto & Luzzatto* By: Attorneys for Applicant		היום This 22 בחודש of אפריל שנה 1998 of the year		
		לשימוש הלישכה		

טופס זה כשהוא מוטבע בחותם לישכת הפטנטים ומושלם במספר ובתאריך ההגשה, הינו אישור להגשת הבקשה שפרטיה רשומים לעיל.
This form, impressed with the Seal of the Patent Office and indicating the number and date of filing, certifies the filing of the application the particulars of which are set out above.

* מחק את המיותר Delete whatever is inapplicable

Ref:4512/97

אפנן QPSK π/4

π/4 QPSK MODULATOR

π/4 QPSK MODULATOR

Field of the Invention

The invention relates to a π/4 shift QPSK modulator for outputting a modulated signal in accordance with inputted digital data bits to use in communications systems, and which is designed for low power consumption and small area.

Background of the Invention

In digital communication systems, a modulator is often used to convert data to analog signals, which can be transmitted e.g. as radio signals or over a telephone line. They functionally operate to shape a data signal and combine the shaped signal with a carrier to provide a modulated data bearing signal. In some applications the shaping function is provided by a FIR (Finite Impulse Response) filter and the resulting shaping is determined by the transfer function of the FIR. The transfer function of the FIR is embodied in an array of coefficients that are to be applied to the inputted data.

π/4-shift QPSK modulators are known in the art and are used in digital communications systems. In the quaternary phase modulation, the signal transmitted in a time interval can be written as $s(t) = A\ (\cos 2\pi f_c t + \theta_k + \varphi)$, where θ_k denotes the message to be sent in said interval, φ is the carrier phase when $t=0$ and in the absence of any phase modulation, and θ_k has eight different values, four at odd times and four at even times, as shown in Fig. 1. In said modulators, the modulated baseband - which, in exponential form, can be written

as $s(t)=e^{j(\theta_k+\varphi)}$ is inputted to a lowpass shaping filter, usually having a symmetric response and typically realized by an FIR digital filter.

In some prior art apparatus, the FIR comprises memory means in which are stored the coefficients a_j for each input signal that is received. For each signal, the corresponding coefficients are retrieved from the memory, The sum-of-products defined by the following formula (1) is then calculated, and all the sums-of-products are added to provide an output. The output can then be converted from digital to analog, as required. Such an apparatus, however, would have to carry out many operations in each unit of time and therefore would have to operate very rapidly.

In the FIR digital filter, the input signal is convoluted with the coefficients which define the filter transfer function. For this purpose, each of the inputs has to be multiplied by a respective coefficient, and the resulting sub-products are to be added together to give an output. The resulting sum-of-products can therefore be expressed as:

$$V_i = \sum_{j=0}^{n-1} a_j u_{i-j} \qquad (1)$$

where V_i is the sum-of-products, i is the time index, u_j are the inputs, a_j are the said coefficients, and j is an integer number which has the values 0 to n-1, wherein n is the number of coefficients.

A sum-of-products multiplier, which performs said operation, is described, e.g., in USP 4,573,136 and in USP 4,953,184. Both USP 4,953,184 and USP 4,573,136 use lookup table means to perform equation (1). However, the number of bits per symbol used as an

addressing to the table is constant. This solution is simple, but leads to big tables which is significantly reduced by the current invention.

A better solution, to which this invention generally refers, is to provide memory means or Look-Up Tables (LUT) which store the values of the V_i. The access to the table at any given time is made to an address which is a combination of all the symbol values being in a FIFO register at that particular time. Fig. 1 illustrates the typical case in which the I-channel and the Q-channel inputs to the filter have five possible values: ±1, ±0.707, and 0, as read on the x-axis of the figure. As seen in Fig. 1, there are four signals at odd times and four at even times. The numbers in Fig. 1 are examples to the I-channel input values resulting from the $\pi/4$ shifted QPSK signal, the Q-channel input will be phase-shifted by $\pi/2$. A memory or LUT required for storing the sums of the products V_i of the input signals for the respective coefficients defining the filter transfer function, has a considerable size. Thus, for example, employing 3 bits to represent said five possible values and for a 6-symbols filter span, and four samples per symbol, the size of memory required is $2^{3 \times 6} \times 4 = 1$ M. Of course, the above numbers are only examples intended to illustrate a general phenomenon that is not limited by them. Note that the conventional QPSK case is much easier for handling, since the I signal and the Q signal can each obtain only two values, and thus one bit is sufficient to represent each value.

Efforts have been made in the art to reduce the size of this memory. Thus, Tetsu Sakata et al., "A New Fully-Digitalized $\pi/4$-Shift QPSK

modulator for Personal Communication Terminals", IEEE, ICUPC 1993, pp. 926-930 propose a reduction of the size of the memory, based on the following considerations. Eight transmission signal points are to be expressed on the orthogonal coordinates system. The $\pi/4$-shift QPSK modulation scheme, as shown in Fig. 1, has four transmission signal points at even times (indicated by squares) and four at odd times (indicated by triangles). The I,Q inputs to the filter, transmitted at even times, have 3 possible values (+1,–1,0) and therefore their amplitude values are described by two bit combinations both in the I-channel and in the Q-channel data. However, the symbols transmitted at odd times have only two possible values (+0.707,–0.707) and a single amplitude, and therefore their amplitude values are described by single bits both in the I-channel and in the Q-channel data. The resulting representation, by two bits and one bit respectively, with reference (as has been noted) to the I-channel, is shown in Fig. 1. Correspondingly, two bits are transmitted on the I-channel and the Q-channel at even times and only one bit is transmitted on both channels at odd times. Since the transmission times shown in Fig. 1 are cyclic, the even times and odd times addresses are, for example:

odd time:

n+5	n+4	n+3	n+2	n+1	n
1	00	0	01	1	11

even time:

n+6	n+5	n+4	n+3	n+2	n+1
01	1	00	0	01	1

This would basically require two separate tables at even and odd times. To overcome this need, in cases when the filter is symmetric in time, the address at odd times can be reversed and the samples can be taken in the reverse order as explained in the above mentioned Tetsu Sakata et al article, in order to use same table at odd and even times. As a result, the size of the memory, for 6 symbols filter span and 4 samples per symbol, is reduced to $2^{1.5^{*}6} \times 4 = 2$ K.

Hereinafter reference will be generally made to 4 samples per symbol and a 6-symbols filter span, but this is only done for illustration purposes and is not a limitation, as different numbers of samples or symbols may be used.

While an important reduction in the size of the memory is thus obtained, it would be desirable to obtain a further reduction, thereby reducing the hardware size and the power requirement. It is therefore a purpose of this invention to provide a $\pi/4$-shift QPSK modulator which has a lower power consumption and a smaller area than any modulator known in the art.

Summary of the Invention

The invention provides a method of $\pi/4$-shift QPSK modulation, which comprises:

defining a transfer function of the modulator by an array of characteristic coefficients;

computing, for each possible inputted symbol sequence, a predetermined number of sample values being the sum of the products of said symbol sequence by said coefficients;

storing said sums-of-products in memory means;

when bits are inputted to the modulator, converting them to symbols, each symbol comprising a number of predetermined samples;

defining the amplitude value of symbols inputted at even times by combinations of two bits and the amplitude value of symbols inputted at odd times by a single bit;

feeding the resulting symbols to shift register means;

addressing each symbol from said register means to said memory means, whereby to identify the corresponding sum-of-products; and

outputting said sum-of-products from said memory means for further processing;

characterized in that it comprises:

successively shifting said symbols in said register means from a Precursor section into a Postcursor section thereof;

counting successive times alternatively divided to odd and even times;

addressing said symbols from said Precursor section to first memory means and from said Postcursor section to second memory means at odd times and from said Precursor section to said second memory means and from said Postcursor section to said first memory means at even times, while further using reversed addressing for Postcursor symbols, in accessing the said memory means

when addressing each symbol, successively incrementing the address indicated by the Precursor section of the register by a number of

storing locations equal to said predetermined number of samples per symbol, and successively decrementing the address indicated by the Postcursor section of the register by a number of storing locations equal to said predetermined number of samples per symbol.

and

adding the sums-of-products outputted from said Precursor section to those outputted from said Postcursor section, whereby to obtain a combined output.

As will explained hereinafter, in different embodiments of the invention said first and second memory means may constitute a single structural unit - one LUT - or two such units. Said Precursor and said Postcursor sections of the register means preferably constitute a single register, but might be two registers, serially connected to one another.

The invention also provides a $\pi/4$-shift QPSK modulator, which comprises:

a bits to symbols conversion unit;

shift register means;

Look-Up Table (LUT) means for storing sums-of-products of symbols by characteristic coefficients defining a modulator transfer function;

address means for converting the symbols from said shift register means to address of said LUT means, whereby to select for each symbol the corresponding sum-of-products values in said LUT means;

and

means for outputting the selected sums-of-products;

characterized in that:

the address means are means also converting the symbols from a Precursor section of said register means at alternate times to a first and a second LUT means and from a Postcursor section of said register means at alternate times to said second and said first LUT means;

the modulator further comprises:

counter means for incrementing or decrementing the address least significant bits from said address means, and for controlling addressing of the symbols from the two register sections to the one or the other of said LUT means, at alternate times; ; and

means for adding the sums-of-products outputted from said first LUT means to those outputted from said second LUT means, whereby to obtain a combined output.

A D/A converter converts the resulting digital combined output to an analog output.

It is seen that, according to the invention, the shift register is conceptually (or, if desired, structurally) divided into a Precursor and a Postcursor as shown in Figs. 2 and 3. If, for example the filter length is 6 symbols , , the Precursor will utilize 3 of those symbols, and the Postcursor will utilize the other 3. The Precursor and Postcursor are computed separately, using time reversed samples for the Postcursor, and the results are added. Thus, assuming a 6-symbol span and 4 samples/symbol, designated by the index i =

1,3,5,7, and using the notation of index $j = 8 - i$, the output is given by:

$$Out[nT+iT/8] = In[(n-2)T] \times H[2T + iT/8] + In[(n-1)T] \times H[T + iT/8] + In[nT] \times H[iT/8] + In[(n+1)T] \times H[(8-i)T/8] + In[(n+2)T] \times H[T + (8-i)T/8] + In[(n+3)T] \times H[2T+(8-i)T/8] =$$

$$In[(n-2)T] \times H[2T + iT/8] + In[(n-1)T] \times H[T + iT/8] + In[nT] \times H[iT/8] + In[(n+1)T] \times H[jT/8] + In[(n+2)T] \times H[T + jT/8] + In[(n+3)T] \times H[2T + jT/8]$$

Wherein n is the symbol index and i is the samples index.

The terms in which the index i occurs define the contribution of the Precursor, and those in which the index j occurs define the contribution of the Postcursor. Addition of the results for all the values of i and j, viz. for both the Precursor and the Postcursor, gives the output resulting from all 6 symbols. It is also clear from the above formula, why the Postcursor uses reversed addressing to the tables, and decreasing sample index.

If the number of symbols is N and N/2 is an odd number, the Precursor has a different address meaning (and length) at even times and at odd times, as follows:

	a	bc	d
odd times	1	01	0

	ef	g	hi
even times	11	0	01

without loss of generality we can assume that the first row above relates to odd times and the second row to even times. Two separate LUTs are needed then for odd and even times. The Postcursor uses the same tables at alternate times, with time reversed samples. The total LUT size for a 6 symbols filter span and 4 samples per symbol is $(2^5+2^4)\times 4 = 192$.

A preferred embodiment of the method of the invention uses symmetry for negated input and expresses negated input by the two's-complement system; and, in a preferred embodiment of the modulator of the invention, which carries out said embodiment of the method, 1 address bit is reduced and 2's complement inverter means are added, thus further reducing the total LUT size to $(2^4+2^3)\times 4 = 96$, at the cost of an extra 2's complement converter, which is a very simple device.

In a further preferred embodiment of the invention, since the LUTs are sparse, the empty entries in one LUT can be used to receive the entries of the other LUT. Since some of the symbols are represented by 2 bits, but obtain only three possible values (00, 01, 11), the extra value, namely 10, which previously holds an empty entry in LUT1, is used to hold values that would otherwise reside in the LUT2.. The two LUT means remain functionally distinct, but are condensed in a single LUT. Two separate accesses must then be provided, but the number of gates can be reduced.

It should be noted that the addresses in the LUTs are identified by two factors: the first (herein indicated by the abbreviation MSB) are the symbols themselves, and is referred to as "high address"; the second is the sample index within the symbol (herein indicated by the abbreviation LSB), and is referred to as "low address" . MSB and LSB together determine the addresses in the LUTs. Referring to the above example, the address of one table is comprised of (a,bc,d), while the address of the second table comprises of (ef,g,hi), where a,d,g hold 1 bit values (0 or 1) and bc,ef,hi hold 2 bit values (00,01,11), while the value 10 is not used. When d equals 0, the address (a,bc,d) uses the empty entry (10,a,bc) in the single LUT. When d equals 1, the address (a,bc,d) uses the empty entry (bc,a,10) in the single LUT.

Brief Description of the Drawings

In the drawings:

- Fig. 1 schematically illustrates the position of the signals in the orthogonal coordinate system;

- Fig. 2 is a block diagram of a QPSK modulator according to an embodiment of the invention;

- Fig. 3 is a block diagram of a QPSK modulator according to another embodiment of the invention;

- Fig. 4 is a block diagram of a QPSK modulator according to still another embodiment of the invention; and

- Figs. 5, 6 and 7 are flow diagrams of the embodiments of Fig. 2, 3 and 4, respectively.

Detailed Description of Preferred Embodiments

It should be noted that the block diagrams of Figs. 2, 3 and 4 are fully adequate structurally to define the modulator of the invention, since all the components thereof, separately considered, are well known in the art and easily provided by skilled persons.

Fig. 2 illustrates one embodiment of the invention in which two tables are used, and all the bits of the Precursor and Postcursor serve as address lines to these tables. Fig. 3 illustrates another embodiment of the invention having two tables in which one bit of the Precursor and Postcursor, serves as a sign bit of the input sequence, and according to which the sign of the output of each table is set. This provides a reduction of the tables size of Fig. 3 by a factor of 2 relative to the tables of Fig. 2.

Fig. 4 illustrates still another embodiment of the invention having only one condensed table holding the values of both of the tables used in Fig. 3.

In the embodiment of Fig. 2, numeral 10 indicates a bit-to-phases converter, into which the bits are fed, as indicated at 51, and from which the converted signals issue with an odd/even representation, as indicated at 52. 11 indicates an I(Q) shift register. Register 11 comprises a Precursor section 53 and a Postcursor section 54, the inputted symbols successively passing through the first and then through the second. The said sections are schematically illustrated for the case of a 6-symbols span. While Precursor and Postcursor could be structurally separated, it is preferred that they constitute a

single register unit, but they are functionally dealt with as separate. The register receives alternatively, at odd and even times, respectively, two-bits and one-bit signal values, and therefore the separation of the register into Precursor and Postcursor sections is not rigid, i.e., in a six symbol register at an even time the Precursor may constitute 4 bits and the Postcursor 5 bits, and at the odd time the Precursor constitutes 5 bits and the Postcursor 4 bits. The size of the said two sections is therefore alternatively changes between 4 and 5 bits. The total size of the register remains the same (9 bits in this example) in all cases. In Fig. 2 the numbers shown in register 11 cells indicate the number of bits in the register for even times.

The register outputs are provided as an address to the look-up tables. In the look-up table, for each register address, a predetermined number of sample values are stored in consecutive locations. An address system is therefore provided, wherein each register's content, the high addresses or most significant bits (MSB), are provided by Muxes and the sample values addresses, least significant bits (LSB), are provided by an oversampling counter 17, to be described later.

Control 25 is a toggling Flip Flop to select between even and odd times. Muxes 12-12', 13-13' and 14-14' read the content of register 11 in odd and even times in a succession established by control 25, in the direct order from the Precursor section 53 and in reverse order from the Postcursor section 54, and determine the high address MSB for access to LUT's 20 and 20'. The LUTs store the sums-of-products, V_i of formula (1), as hereinbefore explained

Both LUTs output the corresponding sums-of-products, indicated above as V_i. An oversampling counter, generally indicated at 16, including a time counter 17, a 2's complement 18, and a Mux 19, determines the low addresses (LSB) for access to the LUTs. Counter 16 sets the LSB address lines of the Precursor section to increment, while the LSB address lines of the Postcursor decrements. Adder 23 sums the data received from tables 20-20', to produce the final output of the filter.

In the embodiment of Fig. 2, the memory of the sums-of-products V_i is shown to obtain address lines from all the bits of the Precursor and Postcursor. As hereinbefore explained, in another embodiment of the invention one of the bits of the Precursor and the Postcursor can serve as a sign of the input sequence, and set the sign of the output of the tables. Such an embodiment is illustrated in Fig. 3, wherein the components that are the same as in Fig, 2 are indicated by the same numerals. In Fig. 3 the output of Muxes 14-14' serve as the above sign bits. If a bit has the negative sign, Muxes 14 and 14' detect it and the sign of the input sequence serving as an address to the LUT is reversed through the 2's complement unit 120, 121, 122, and 123, selected through Muxes 115, 116, 117 and 118. Muxes 14 and 14' also transfer the negative sign to Muxes 21 and 21', whereby to cause the reversal of the sign of the LUT outputs through the 2's complement unit 24 and 24'.

In the embodiments of Figs. 2 and 3, the memory of the sums-of-products V_i comprises two LUTs, 20 and 20'. As hereinbefore explained, in still another embodiment of the invention only one LUT may be used for odd and even times, which LUT combines the entries of both the LUTs of Fig. 3. Such an embodiment is illustrated in Fig. 4, wherein the components that are the same as in Fig. 3 are indicated by the same numerals. A single LUT 40 is provided, and a modification with respect to the embodiment of Fig. 3 is introduced as to the access to the contents of LUT 40. Once again, the addresses are defined by high addresses MSB and low addresses LSB. However, when a new symbol is inputted, two accesses are performed to Table 40, one for precursor value, then for postcursor value. Unit 126 is a toggle Flip-Flop which designates first/second access, and also indicates the same to Mux 19 of the oversampling counter, for selecting incrementing/decrementing of the same as for purposes as hereinbefore explained. Control 25 designates odd/even times, and X-OR 127 provides proper selections accordingly to Muxes 42, 43, and 44. The first access provides the value previously stored in the Table 20 of Fig. 3. This is done by the fact that Muxes 42, 43 and 44 select the outputs of Muxes 115, 116 and 14 as the MSB of the address to Table 40 for the first access. The second access provides the value previously stored in Table 20' of Fig. 3. This is done by the fact that Muxes 42, 43 and 44 select the outputs of Muxes 117, 118 and 14' as the MSB of the address to Table 40 for the second access. Since table 40 contains values of both table 20 and table 20' of Fig. 3, through the mechanism explained above, Muxes 42, 43 and 44, cause the selection of previously empty places of LUT 20 of Fig. 3, now containing values

previously stored in table 20', to be addressed. Element 125 is a logical converter that converts addresses of table 20' of Fig.3 to empty addresses in table 20 through the use of the value 10 which is not used for 2 bits symbols as explained above. Mux 44 also transfers the sign, received from Mux 14 or 14', to Mux 45, which has the same function as Muxes 21 and 21' in the preceding embodiment. The 2's complement unit 46 serves the same function as the similar units 24 and 24' of Fig. 3. The outputs of Mux 45 are added by adder 47 for the said two accesses to table 40, from which the final output of the modulator issues, as indicated at 48. Zeroing means 49 are provided for zeroing the adder at the beginning of the operation.

Fig. 5 is a flow diagram of the operation of a modulator using two LUTs, such as illustrated in Fig. 2. In the flow chart, only the (I) channel is described, since the (Q) channel is parallel and identical. The diagram refers to a 6-symbol example. The various blocks as indicated in Fig. 5 perform the following operations:

Flow Chart of Fig. 5

300 - Start at even timing.

301 - Enter next symbol into the register, one bit at odd times, two bits at even times, withdraw old symbol.

302 - Sample counter = 1.

303 - Is the time even?

304 - Symbols 1, 2, 3 are used as high address lines for table 2.

305 - Symbols 4, 5, 6 are used as high address lines for table 1.

315 - Symbols 1, 2, 3 are used as high address lines for table 1.

314 - Symbols 4, 5, 6 are used as high address lines for table 2.

313 - The sample counter outputs are used as low address lines for table 1, and after inversion, as low address lines for table 2.

312 - Add result from table 1 (data from address consisting of high and low address lines), with result from table 2 (data from address consisting of high and low address lines). Output the result.

311 - Is it the last sample of the symbol?

316 - Increment the counter of the samples in the symbol.

Fig. 6 is a flow diagram of the operation of a modulator using two LUTs, such as illustrated in Fig. 3. In the flow charts, only the (I) channel is described, since the (Q) channel is parallel and identical. The diagram refers to a 6-symbol example. The various blocks as indicated in Fig. 6 perform the following operations:

Flow Chart of Fig. 6

101 - Start at even timing.

102 - Enter next symbol into the register, one bit at odd times, two bits at even times, withdraw old symbol.

103 - Sample counter = 1.

104 - Is the time even?

105 - Symbols 1, 2, 3 are used as high address lines in table 2, while symbol 1 is used as the sign.

106 - Symbols 4, 5, 6 are used as high address lines in table 1, while symbol 5 is used as the sign.

107 - The sample counter outputs are used as low address lines in table 2, and after inversion, as low address lines in table 1.

108 - Add result from table 1 (address consisting of high and low address lines), with result from table 2 (address consisting of high and low address lines), while the sign is the symbol sign. Output the result.

109 - Is it the last sample of the symbol?

110 - Increment the symbol's sample counter.

116 - Symbols 1, 2, 3 are used as high address lines in table 1, and symbol 2 is used as the sign.

115 - Symbols 4, 5, 6 are used as high address lines in table 2, while symbol 4 is used as the sign.

114 - The sample counter is used as low address lines in table 1, and after inversion, as low address lines in table 2.

113 - Add result from table 1 (address consisting of high and low address lines), with result from table 2 (address consisting of high and low address lines), while the sign is the symbol sign. Output the result.

111 - Increment the counter of the samples in the symbol.

112 - Is it the last sample of the symbol?

Fig. 7 is a flow diagram similar to that of Figs. 5 and 6, but illustrating the operation of a modulator using a single LUT, as illustrated in Fig. 4. In this case as well, only the (I) channel is described, since the (Q) channel is parallel and identical, the diagram refers to a 6-symbol example, and only the (I) channel is described.

Flow chart of Fig. 7

200 - Start at even timing.

201 - Enter next symbol into the register, one bit at odd times, two bits at even times, withdraw old symbol.

202 - Sample counter = 1.

203 - Is the time even?

204 - Symbols 1, 2, 3 are used as high address lines in the table for the first access, while symbol 1 is used as the sign. Convert symbols 2 and 3 to 4 address lines by a conversion exploiting the empty locations of original table 1.

205 - Symbols 4, 5, 6 are used as high address lines in the table, for a second access to the table, while symbol 5 is used as the sign.

206 - The sample counter is used as low address lines in the table in first access, and after inversion as low address lines in a second access to the table.

207 - Add result of first access to the table from address consisting of high and low address lines, with result of second access to the table from address consisting of low and high address signs, while the sign of each result is determined by the corresponding sign symbol. Output the result.

208 - Is it the last sample?

209 - Increment the counter of the samples in the symbol.

215 - Symbols 1, 2, 3 are used as high address lines to the table, while symbol 2 is the sign.

214 - Symbols 4, 5, 6 are used as high address lines to the table in a second access to the table, while symbol 4 is used as the sign.

Convert the symbols 5 and 6 into 4 address lines by a conversion exploiting the empty locations of the original table 1.

213 - The sample counter is used as low address lines in the table in first access, and after inversion as low address lines in a second access to the table.

212 - Add result of first access to the table from address consisting of high and low address lines, with result of second access to the table from address consisting of low and high address lines, while the sign of each result is determined by the corresponding sign symbol. Output the result.

210 - Increment the symbol sample counter.

211 - Is it the last sample of the symbol?

As has been said the size of the LUTs according to this invention is considerably smaller than that of the prior art LUTs. The following examples illustrate this fact. S is the size of the LUT or the combined size of LUT1 and LUT2, Nsy the number of symbols spanned, and Nsa the number of samples per symbol.

In the embodiment of Fig. 2:

if Nsy is a multiple of 4, $S = (2^{[1.5\times Nsy]})\times Nsa$

if Nsy is not a multiple of 4, $S = (2^{1.5\times Nsy/2+0.5} + 2^{1.5\times Nsy/2-0.5}) \times Nsa.$

In the embodiment of Fig. 3:

if Nsy is a multiple of 4, $S = (2^{[1.5\times Nsy/2-1]})\times Nsa$

if Nsy is not a multiple of 4, $S = (2^{[1.5\times Nsy/2-0.5]} + 2^{[1.5\times Nsy/2-1.5]}) \times Nsa.$

In the embodiment of Fig. 4, if Nsy is not a multiple of 4, $S = (2^{[1.5\times Nsy/2-0.5]})\times Nsa$.

If Nsy is a multiple of 4 tables 20 and 20' of Fig. 3 are already identical, and thus to incorporate them in a single table, takes no effort, only accessing the same table with twice the speed.

While some embodiments of the invention have been described by way of illustration, it will be apparent that the invention can be carried into practice with many modifications, variations and adaptations, and with the use of numerous equivalents or alternative solutions that are within the scope of persons skilled in the art, without departing from the spirit of the invention or exceeding the scope of the claims.

CLAIMS

1. Method of $\pi/4$-shift QPSK modulation, which comprises:

defining a transfer function of the modulator by an array of characteristic coefficients;

computing, for each possible inputted symbol sequence, a predetermined number of sample values being the sum of the products of said symbol sequence by said coefficients;

storing said sums-of-products in memory means;

when bits are inputted to the modulator, converting them to symbols, each symbol comprising a number of predetermined samples;

defining the amplitude value of symbols inputted at odd times by combinations of two bits and the amplitude value of symbols inputted at even times by a single bit;

feeding the resulting symbols to shift register means;

addressing each symbol from said register means to said memory means, whereby to identify the corresponding sum-of-products; and

outputting said sum-of-products from said memory means for further processing;

characterized in that it comprises:

successively shifting said symbols in said register means from a Precursor section into a Postcursor section thereof;

counting successive times alternatively divided to odd and even times;

addressing said symbols from said Precursor section to first memory means and from said Postcursor section to second memory means at

odd times and from said Precursor section to said second memory means and from said Postcursor section to said first memory means at even times, while further using reversed addressing for Postcursor symbols, in accessing the said memory means;

when addressing each symbol, successively incrementing the address indicated by the Precursor section of the register by a number of storing locations equal to said predetermined number of samples, and successively decrementing the address indicated by the Postcursor section of the register by a number of storing locations equal to said predetermined number of samples.

2. Method according to claim 1, wherein the first and second memory means are integrated in a single Look-Up Table.

3. $\pi/4$-shift QPSK modulator, which comprises:

a bits to symbols conversion unit;

shift register means;

Look-Up Table (LUT) means for storing sums-of-products of symbols by characteristic coefficients defining a modulator transfer function;

address means for converting the symbols from said shift register means to address of said LUT means, whereby to select for each symbol the corresponding sum-of-products values in said LUT means; and

means for outputting the selected sums-of-products;

characterized in that:

the address means are means also converting the symbols from a Precursor section of said register means at alternate times to a first

and second LUT means and from a Postcursor section of said register means at alternate times to said second and first LUT means correspondingly;

the modulator further comprises:

counter means for incrementing/decrementing the address least significant bits from said address means, and for controlling addressing of the symbols from the two register sections to the one or the other of said LUT means, at alternate times;

means for adding the sums-of-products outputted from said first LUT means to those outputted from said second LUT means, whereby to obtain a combined output; and

Digital to analog means for converterting the combined output to an analog output.

4. Modulator according to claim 3, wherein the LUT means are constituted by two separate LUTs, the first LUT is accessed at odd times by the Precursor section address, and at even times by the Postcursor section address, and the second LUT is accessed at odd times by the Postcursor section address, and at even times by the Precursor section address.

5. Modulator according to claim 3, comprising a single LUT and means for using the empty entries in one LUT are used to store the entries that would otherwise be stored in another LUT.

6. Modulator according to claim 3, wherein the addresses in the LUT means are identified by a high address factor (MSB), which is the

inputted symbol, and by a low address factor (LSB), which is the order of sampling within the symbol.

7. Modulator according to claim 6, wherein the address means comprise Muxes and gates for providing the MSB address and an oversampling counter for providing the LSB address.

8. Method according to claim 1, wherein the shift register means is *n* symbols register, *n* being an even number, the Precursor utilizes n/2 of said symbols and the Postcursor utilizes the remaining n/2 symbols, addition of the Precursor sum-of-products and Postcursor sum-of-products providing the output.

9. Method according to claim 1, comprising expressing negated input by one of the bits of the Precursor and one of the bits of the Postcursor sections, and when a bit indicating negative input is detected, reversing the sign of the input sequence addressing the memory means by applying 2's complement operation on the input sequence, and further reversing the sign of the LUT outputs by applying 2's complement operation on the output from the memory means.

לוצאטו את לוצאטו
LUZZATTO & LUZZATTO
By: ע"י:

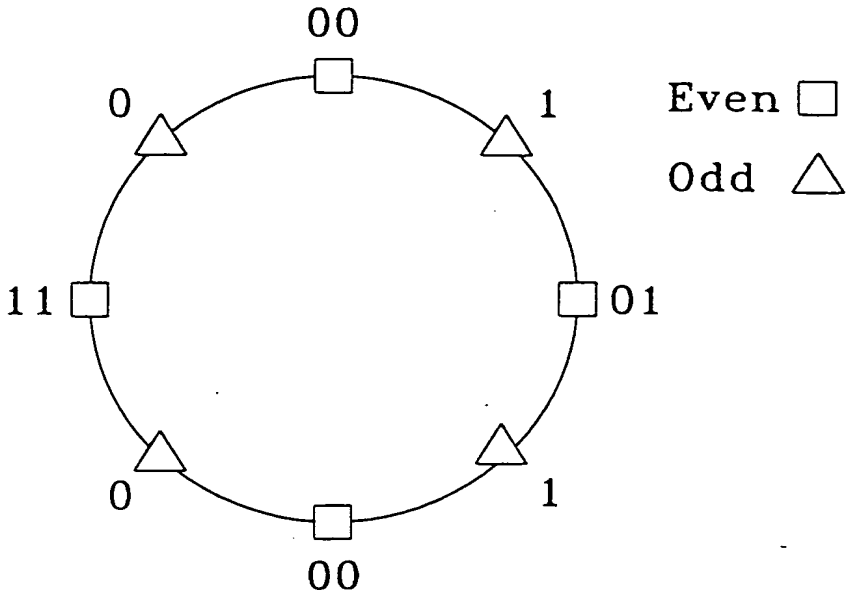

Fig. 1

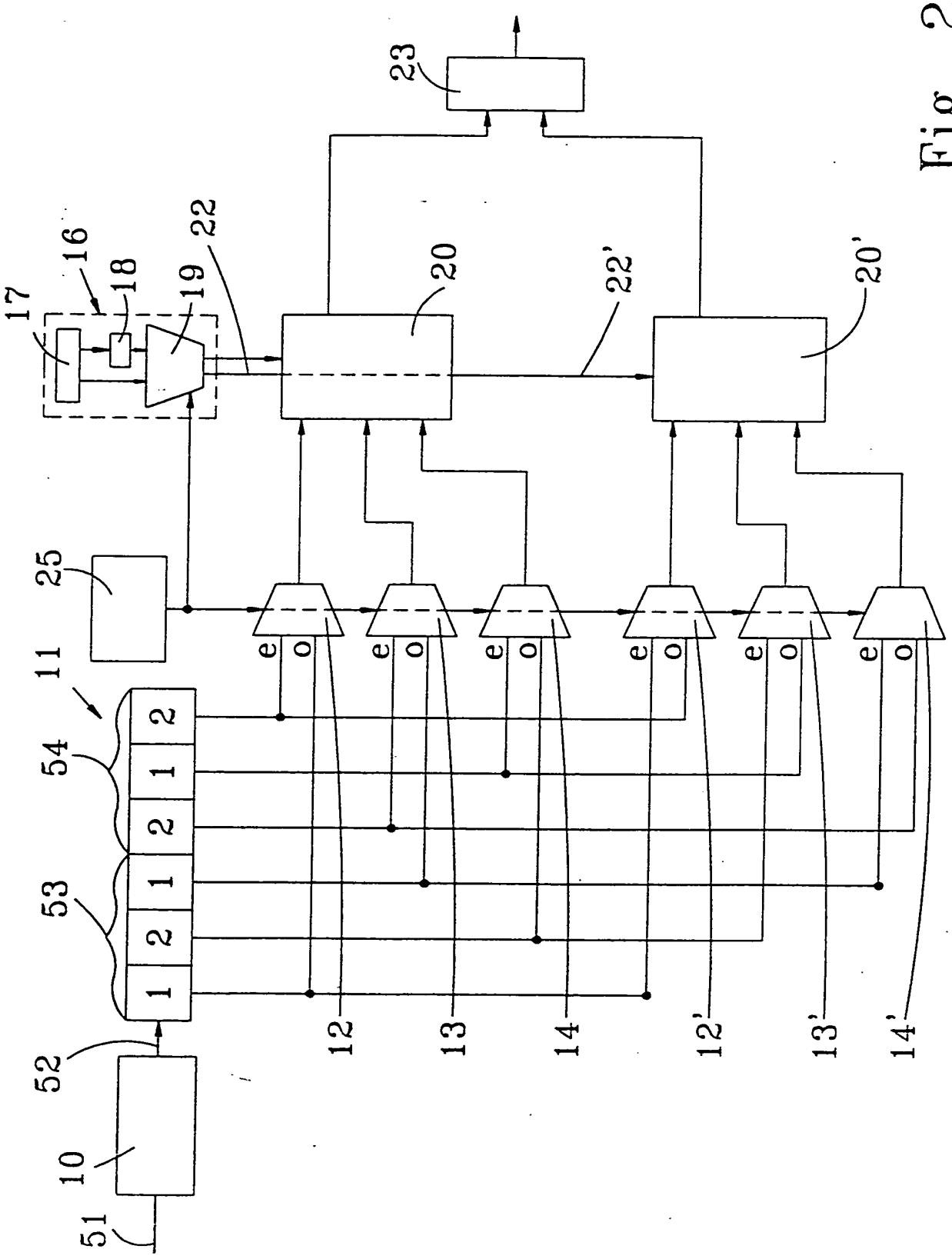

Fig. 2

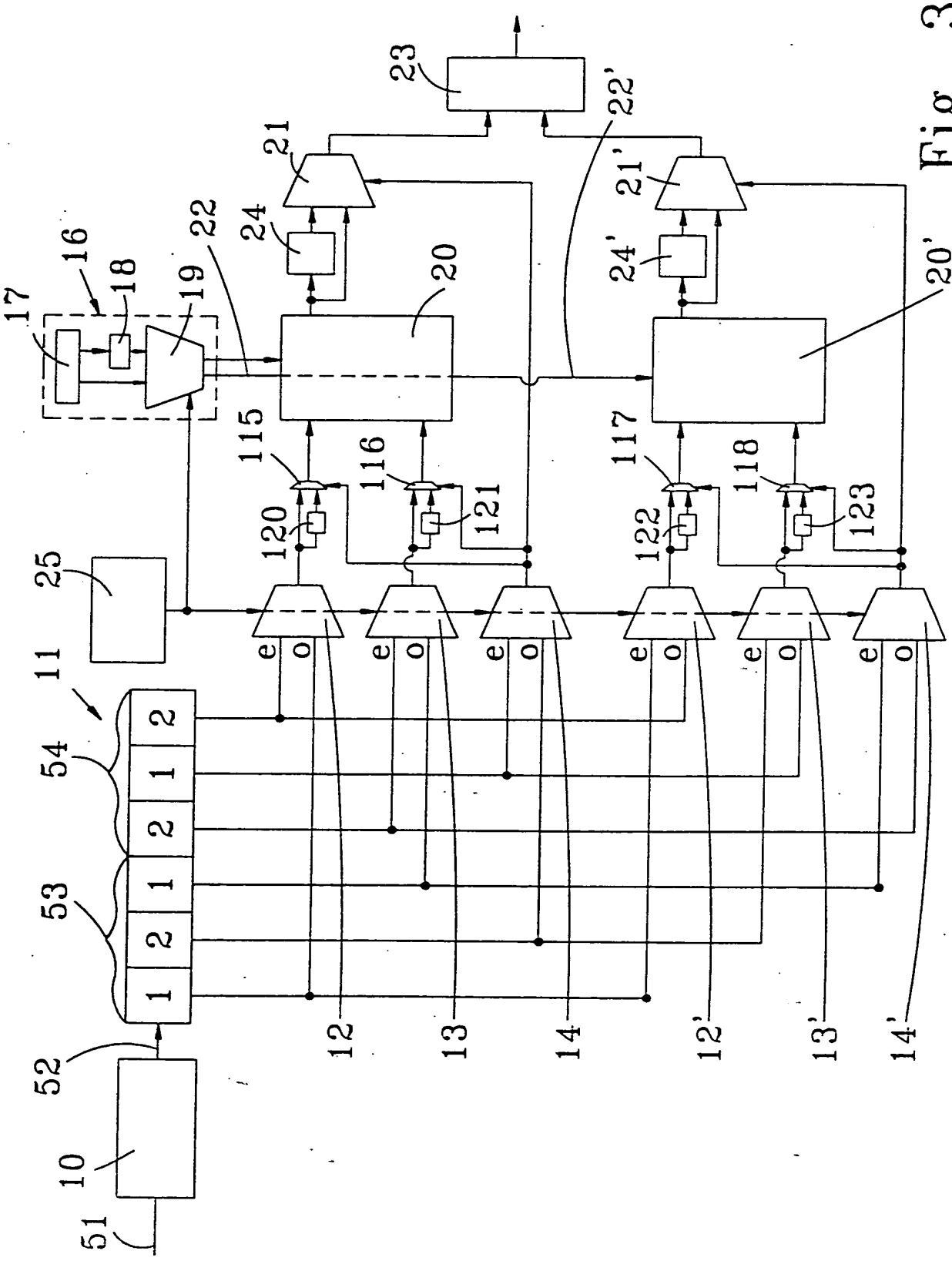
23
22'
21
21'
24
24'
20
20'
16
18
19
22
17
115
116
117
118
120
121
122
123
25
11
54
53
52
10
51
12
13
14
12'
13'
14'
e
o
2
1

Fig. 3

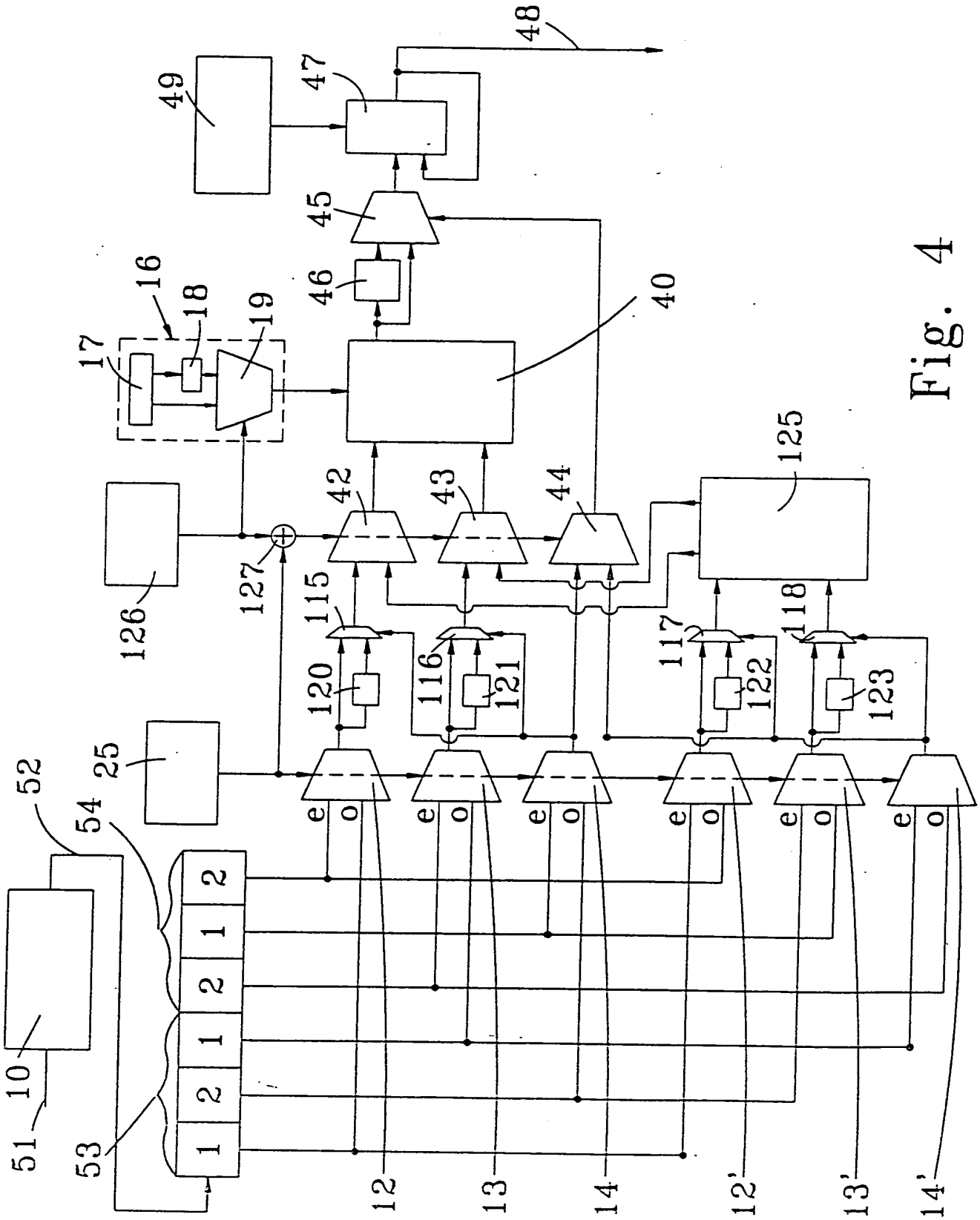
48
47
49
45
46
40
16
18
19
17
125
42
43
44
126
127
115
118
117
116
120
121
122
123
25
52
54
e
o
2
1
2
1
2
1
10
51
53
12
13
14
12'
13'
14'

Fig. 4

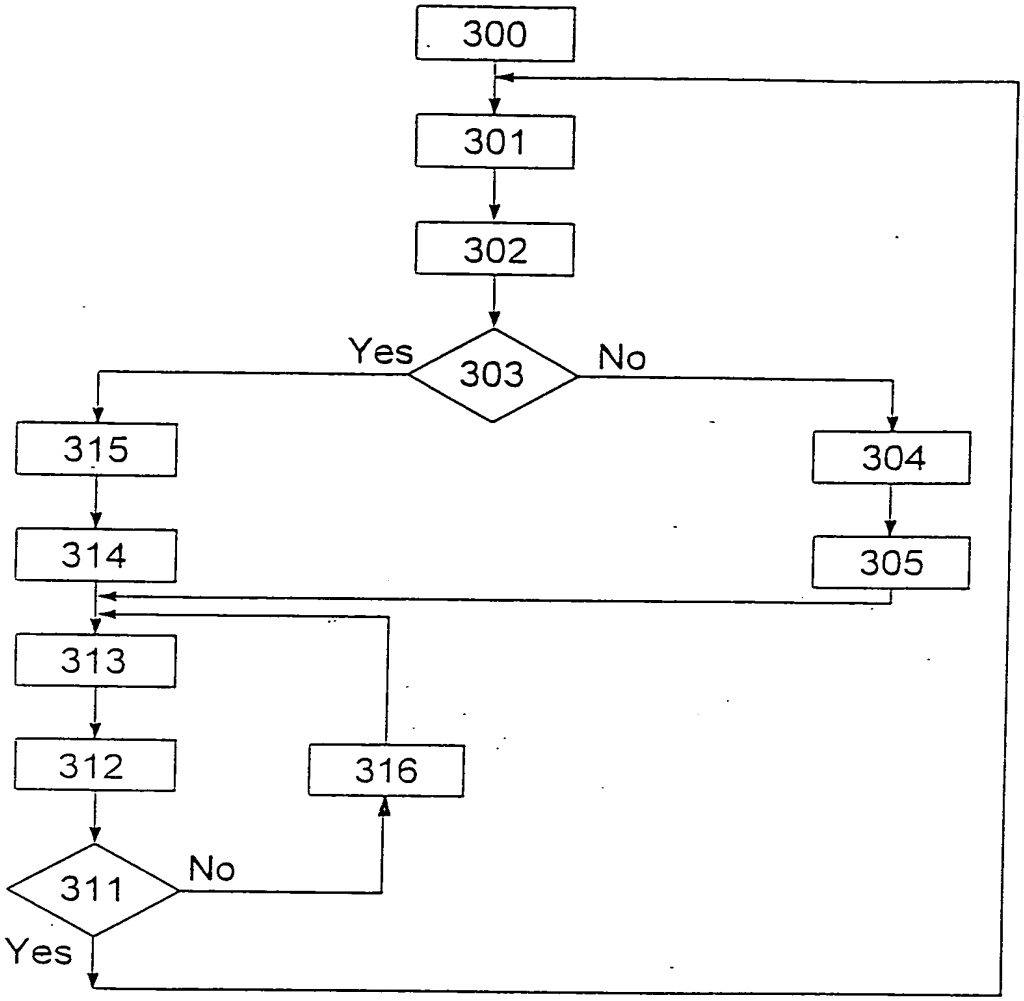

Fig. 5

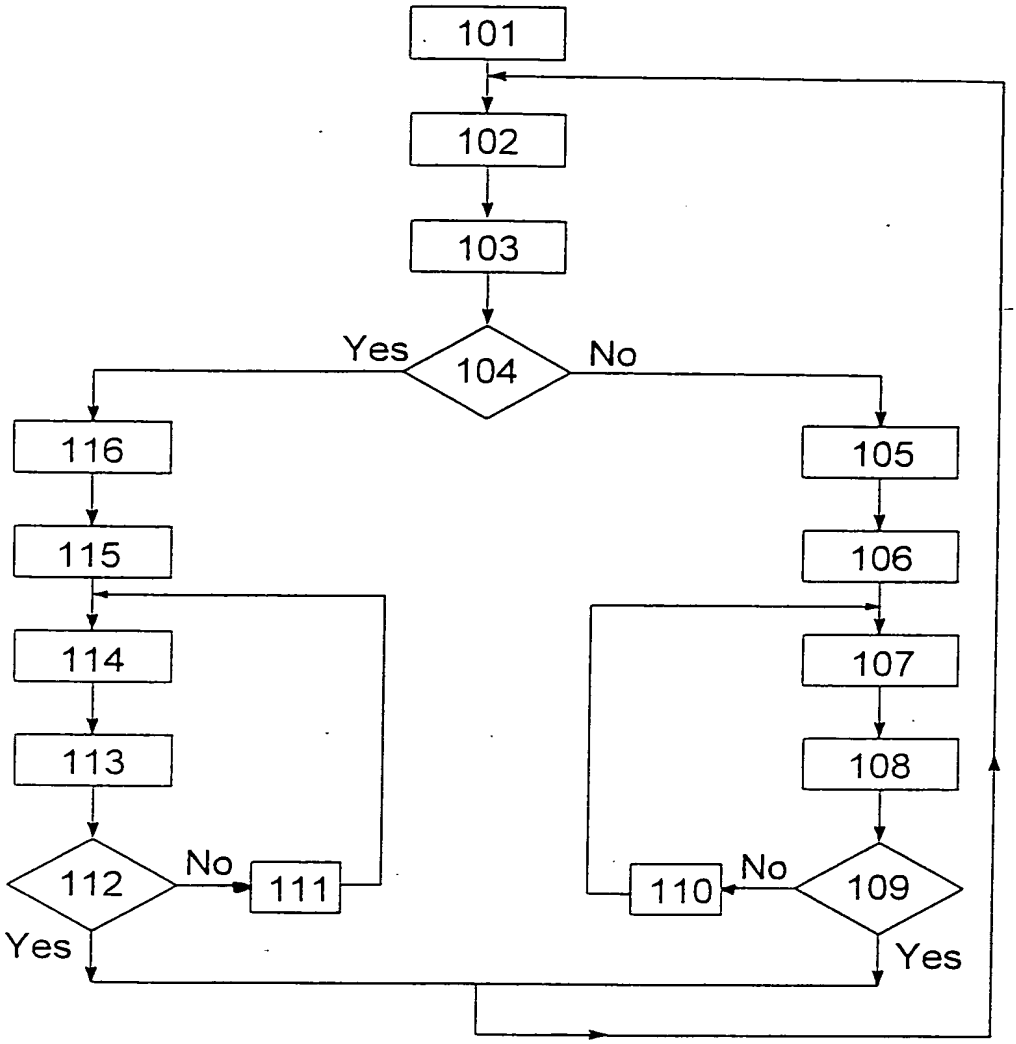
101
102
103
Yes
104
No
116
105
115
106
114
107
113
108
112
No
111
110
No
109
Yes
Yes

Fig. 6

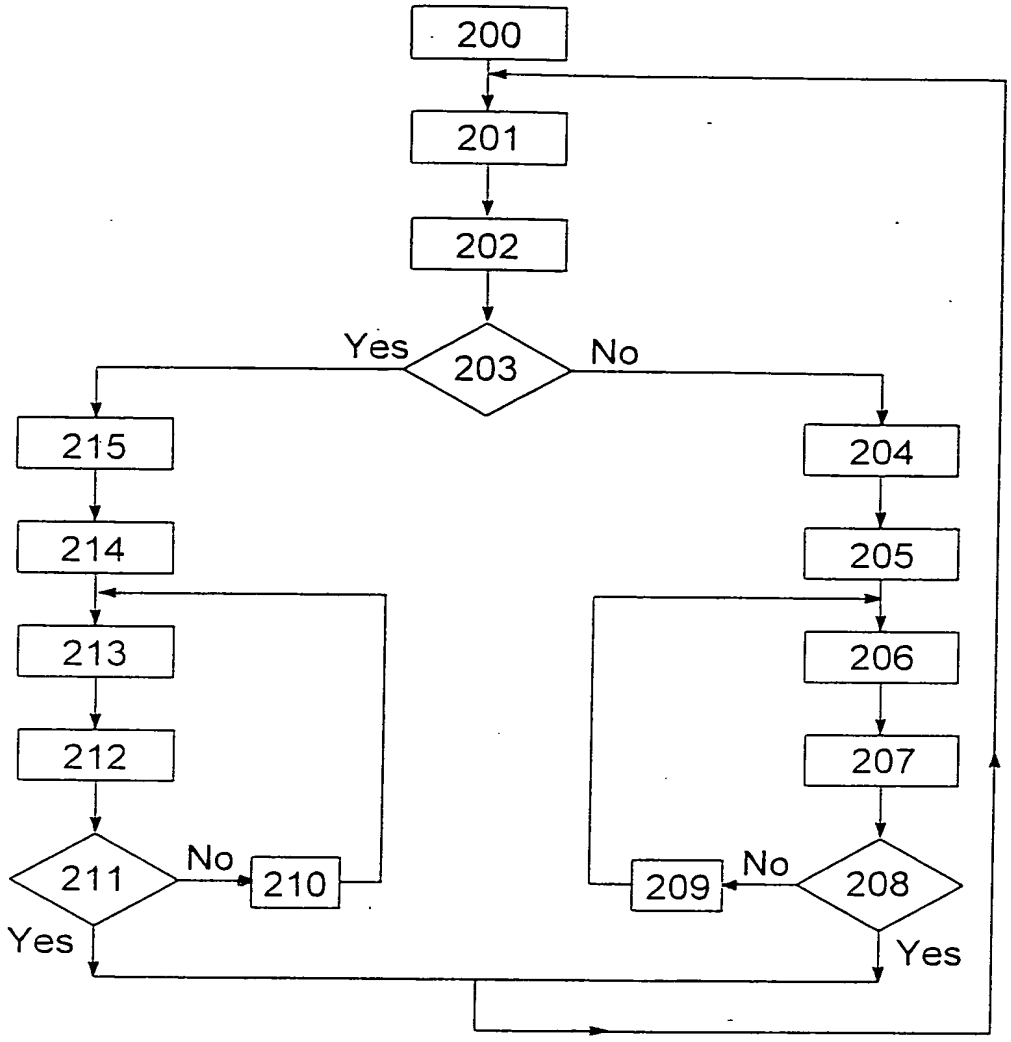
200
201
202
Yes
203
No
215
214
213
212
211
No
210
Yes
204
205
206
207
209
No
208
Yes

Fig. 7